**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently amended) A computer-implemented method for controlling access to a data object stored in a non-volatile memory, the data object having an identifier (ID), the method comprising:

checking, before accessing the data object, whether the ID is contained in a lock object and the ID is associated with a storage location; ~~and~~

accessing the data object, if the ID is not contained in the lock object or if the ID is not yet associated with a storage ~~location.~~ location;

determining whether a transactional lock has been successfully set on the data object;

determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and

granting read/write access to the data object based on the permanent lock not being set on the data object.

2. (Previously Presented) The method of claim 6, comprising:

deleting the ID from the second lock object, if the ID is not yet associated with a storage location.

3. (Original) The method of claim 1, wherein the lock object comprises a table, having a column for the ID and a column for a link to the storage location associated with the ID.

4. (Original) The method of claim 2, wherein
the lock object comprises a table, having a column for the ID and a column for a link to the storage location associated with the ID.

5. (Original) The method of claim 1, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

6. (Original) The method of claim 1, further comprising:

before performing the check, storing the ID in a second lock object, which is stored in a volatile storage means.

7. (Original) The method of claim 6, further comprising:

checking, whether the ID has been successfully stored in the second lock object before accessing the data object and, if the ID has not been successfully stored in the second lock object, not accessing the data object.

8. (Currently amended) A computer system for controlling access to a data object having an identifier (ID), the system comprising:

memory having program instructions;

storage means for storing data;

at least one processor to execute the program instructions to perform operations comprising:

checking, before accessing the data object, whether the ID is contained in a lock object and the ID is associated with a storage location; and

accessing the data object, if the ID is not contained in the lock object or if the ID is not yet associated with a storage ~~location.~~ location;

determining whether a transactional lock has been successfully set on the data object;

determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and

granting read/write access to the data object based on the permanent lock not being set on the data object.

9. (Previously Presented) The computer system of claim 13, further comprising:

deleting the ID from the second lock object, if the ID is not yet associated with a storage location.

10. (Original) The computer system of claim 8, wherein the lock object comprises a table, having a column for the ID and a column for a link to the storage location associated with the ID.

11. (Original) The computer system of claim 9, wherein the lock object comprises a table, having a column for the ID and a column for a link to the storage location associated with the ID.

12. (Original) The computer system of claim 8, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

13. (Original) The computer system of claim 8, wherein before performing the check, storing the ID in a second lock object, which is stored in a volatile storage means.

14. (Original) The computer system of claim 13, wherein checking, whether the ID has been successfully stored in the second lock object before accessing the data object and, if the ID has not been successfully stored in the second lock object, not accessing the data object.

15. (Currently amended) A computer-readable medium comprising instructions for controlling access to a data object having an identifier (ID), the medium comprising instructions for:

checking, before accessing the data object, whether the ID is contained in a lock object and the ID is associated with a storage location; and

accessing the data object, if the ID is not contained in the lock object or if the ID is not yet associated with a storage ~~location.~~ location;

determining whether a transactional lock has been successfully set on the data object;

determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and

granting read/write access to the data object based on the permanent lock not being set on the data object.

16. (Previously Presented) The medium of claim 20, comprising:

deleting the ID from the second lock object, if the ID is not yet associated with a storage location.

17. (Original) The medium of claim 15, wherein the lock object comprises a table, having a column for the ID and a column for a link to the storage location associated with the ID.

18. (Original) The medium of claim 16, wherein the lock object comprises a table, having a column for the ID and a column for a link to the storage location associated with the ID.

19. (Original) The medium of claim 15, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

20. (Original) The medium of claim 15, wherein before performing the check, storing the ID in a second lock object, which is stored in a volatile storage means.

21. (Original) The medium of claim 20, wherein checking, whether the ID has been successfully stored in the second lock object before accessing the data object and, if the ID has not been successfully stored in the second lock object, not accessing the data object.

22. (Currently amended) A memory for storing data for access by a process being executed by a processor, the memory comprising:

a structure for controlling access to a data object having an identifier (ID), the structure comprising a first lock object, storing the ID of the data object and a link to a storage location where the data object is stored, and a second lock object storing the ID of the data ~~object.~~ object; and

instructions for:

determining whether a transactional lock has been successfully set on the data object;

determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and

granting read/write access to the data object based on the permanent lock not being set on the data object.

23. (Original) The memory of claim 22, wherein the first lock object comprises a table, having a column for the ID and a column for the link to a storage location where the data object is stored.

24. (Original) The memory of claim 22, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

25. (Original) The memory of claim 23, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

26. (Original) The memory of claim 22, wherein the first and second lock objects are created by a data moving or data archiving process.

27. (Previously presented) A computer-implemented method for controlling access to a data object having an identifier (ID), the method comprising:

selecting the data object having the ID;

determining whether a transactional lock has been successfully set on the data object;

determining whether a permanent lock has been set on the data object based on the transactional lock being successfully set on the data object;

granting read/write access to the data object based on the permanent lock not being set on the data object; and

deleting the transactional lock.

28. (Currently amended) A computer-implemented method for controlling access to a data object having an identifier (ID), the method comprising:

selecting the data object;

checking, before accessing the data object, whether the ID is contained in a permanent lock ~~object, wherein the data object is scheduled to be archived if the data object's ID is contained in the permanent lock~~ object;

archiving the data object if the data object's ID is contained in the permanent lock object; and

granting access to the data object if the ID is not contained in the permanent lock object.